AD A091608

(6) STEEPEST EDGE
ALGORITHMS IN LINEAR AND NONLINEAR PROGRAMMING

(14) Final Report 80-1

LEVEL II

by

(10) DONALD GOLDFARB

(11) September 1980

(12) (8)

U.S. ARMY RESEARCH OFFICE
GRANT NUMBER (15) DAAG-29-77-G-0114

The City College of The City University
of New York.
(3) Department of Computer Science
New York, N. Y. 10031

412 028

DTIC
ELECTE
NOV 1 2 1980
B

412 028

8011 03 103

DDC FILE COPY

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 80-1	2. GOVT ACCESSION NO. AD-A091 608	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Steepest Edge Algorithms for Linear and Nonlinear Programming		5. TYPE OF REPORT & PERIOD COVERED Final Report 1 June 1977 - 31 July 1980
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) DONALD GOLDFARB		8. CONTRACT OR GRANT NUMBER(s) Grant DAA29-77-G-0114
9. PERFORMING ORGANIZATION NAME AND ADDRESS The City College of the City University of New York Dept. of Computer Science New York, N. Y. 10031		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Research Office Post Office Box 12211 Research Triangle Park, NC 27709		12. REPORT DATE September, 1980
		13. NUMBER OF PAGES 5
14. MONITORING AGENCY NAME & ADDRESS(If different from Controlling Office) Dept. of the Navy Office of Naval Research Resident Representative 715 Broadway (5th Floor) New York, N. Y. 10003		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

NA

18. SUPPLEMENTARY NOTES

The view, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Optimization, Linear Programming, Quadratic Programming, Nonlinear Programming, Steepest Edge Algorithm, Network Flows, Maximum Flow, Ellipsoid Method, Computational Complexity, Steplength Algorithms, Negative Curvature, LU factorization, Sparse Matrices

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Several algorithms in linear and nonlinear programming have been developed and analyzed. A worst-case analysis of the steepest edge simplex method showed it to have exponential time-complexity. This algorithm was specialized for solving minimum cost network flow problems and a partial pricing variant was developed. After extensive testing on randomly generated large problems the method was found to be inferior to efficiently coded partial pricing

DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Block 20

variants of the "standard" simplex method except for the max flow problem and the problem of finding a feasible flow. On the max flow problem the algorithm was compared with the Edmonds-Karp and Dinic algorithms and found to be superior. In quadratic programming, the use of the steepest edge (face) criterion for dropping constraints was found to be helpful. Dual methods were found to be even more efficient and their use in recursive quadratic programming algorithms for nonlinear programming problems is recommended. A numerically stable ellipsoid algorithm for linear programming was developed and analyzed. At present the main promise that this method holds is as a powerful theoretical tool. Several minimization algorithms for taking advantage of negative curvature were developed as was a curvilinear steplength algorithm which ensures convergence to a positive semidefinite point.
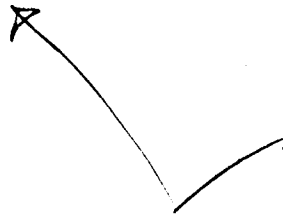
Accession For	
NTIS GRA&I	✔
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	

A. PROBLEM STATEMENT

Although we originally only proposed to study steepest edge algorithms
in linear and nonlinear programming, exciting new developments and
and our own scientific curiousity led us to broaden our research into
other types of algorithms in linear and nonlinear programming as well.
Our researches, however, remained limited to a study of algorithms
in these two areas.

In linear programming a number of problems were studied including
(i) the computational complexity of the steepest edge simplex algorithm,
(ii) the ellipsoid algorithm of Shor and Judin and Nemirovskii, which
was used by Khachiyan to show that linear programs could be solved
in polynomial time, and (iii) the preservation of sparsity when up-
dating the LU factorization of the basis matrix in the simplex method.

For the subclass of linear programming problems which are subsumed
under the category of minimum cost   network flow problems, we
studied (i) the development of efficient steepest edge simplex algorithms
for these problems, (ii) how to choose appropriate data structures
to implement these algorithms, (iii) the development of specialized
algorithms for the max flow problem and (iv) the avoidance of de-
generate pivots.

In the closely related subject of quadratic programming we studied
(i) steepest edge (i.e., face) algorithms, and (ii) dual and primal-
dual algorithms.

In the area of nonlinear programming we studied the problems of (i)
developing algorithms which can take advantage of negative curvature
in the objective function, and (ii) how to determine the steplength
to take along a curvilinear path in a region where the function
being minimized is nonconvex.

B. SUMMARY OF RESEARCH RESULTS
1. Linear Programming Algorithms
(i) Computational Complexity of the Steepest Edge Simplex Algoithm.

We have investigated the computational complexity of the steepest-
edge simplex algorithm for general linear programming problems.
We have shown how to construct linear programs whose feasible
regions are combinatorially equivalent to hypercubes of arbitrary
dimension n for which the steepest edge simplex algorithm passes
through all 2^n vertices. This "worst-case" analysis extends the
well-known analogous result of Klee and Minty for the standard
pivot section rule.

(ii) The Ellipsoid Method

A rigorous proof has been given of the fact that among all ellipsoids
that contain a given half-ellipsoid the one constructed by the

ellipsoid method is of minimum volume. The deep-cut version of
the ellipsoid method was analyzed for a special class of problems
and it was shown that this version is only four to five times
as fast as the Shor-Khachiyan version. It was also demonstrated
that although the volumes of the ellipsoids generated converge
to zero, the points (i.e. centers of the ellipsoids) need not
converge to a feasible point if the feasible set is not full-dimensional
Also, in the infeasible case it was shown that infeasibility can
go undetected.

A survey of past and current work on the ellipsoid method was
written in collaboration with Profs. Robert Bland and Michael
Todd of Cornell University, and a numerically stable algorithm
was developed for solving linear programming problems in
collaboration with Professor Michael Todd. Our implementation
updates at each iteration the Cholesky factors of the symmetric
positive definite matrix which defines the ellipsoids in a way
which ensures numerical stability. By using so-called "surrogate"
cuts and objective function cuts and by refining both lower and
upper bounds on the optimal value of the objective function, our
algorithm should be able to terminate with an indication of in-
feasibility or with a provably good feasible solution in a moderate
number of iterations. The method was coded and tested on a small
set of problems. Unfortunately, the results obtained were not
very promising even though all sorts of devices to accelerate
convergence were employed.

(iii) Updating of Sparse LU Factorization of an LP Basis Matrix:

A new way to preserve as much sparsity as possible when updating
the LU factorization of a basis matrix (in the simplex method,
for example) has been developed. By proper choice of row and
column permutations it produces a block upper triangular matrix
with small diagonal blocks which is then triangularized by the
Bartels-Golub algorithm.

2. Minimum Cost Network Flow Algorithms
   (i) Steepest Edge Simplex Algorithms./(ii) Implementation.

A steepest edge simplex algorithm for network flow problems has
been developed which updates the number of arcs in each nonbasic
cycle, (i.e., flow augmenting path), after each simplex pivot
from properties of the basic tree. A partial pricing variant
of this network steepest edge simplex method has also been
developed. A network simplex code using list structures for
storing the basic tree and including several partial pricing
strategies as options has been written in FORTRAN.

A novel feature of the code is the use of a postorder thread
to represent the basis tree. This code was subjected to extensive

computational testing on thirty randomly generated large problems
(thousands of nodes and tens of thousands of arcs) - to determine
the most efficient level of partial pricing.

Our principal finding is that the decrease in iterations that
results from using the steepest edge column selection rule in either
a full or partial pricing algorithm does not compensate sufficiently
for the increase in work per iteration over that required by the
"minimum reduced cost" rule on most network flow problems. Exceptions
to this are the max flow problem and the problem of finding a feasible
flow.

(iii) Max Flow Algorithm:

A partial pricing variant of the steepest edge simplex algorithm for
minimum cost network flow problems was found to be very efficient
for solving max flow problems. A FORTRAN program for solving max
flow problems by the steepest edge simplex method was developed
in collaboration with Dr. Michael Grigoriadis on the basis of
his RNET code. This was compared computationally with our own
FORTRAN codes for the Edmonds-Karp algorithm and Dinic's algorithm.
Our results indicate that our method is superior to the latter
method and that both the steepest edge and Dinic algorithms are
vastly superior to the Edmonds-Karp algorithm.

(iv) Avoiding Degenerate Pivots.

Two techniques for avoiding degenerate pivots in network flow
problems have been developed, but not yet computationally tested.
They are based upon the use of special list structures.

3. Quadratic Programming Algorithms.
    (i) Steepest Edge (Face) Algorithms.

The steepest edge(face)criterion for dropping constraints from
an active set in primal quadratic programming algorithms has been
implemented in a FORTRAN code.

Extensive computational testing indicated that use of the steepest
face criterion results in fewer steps and basis changes. However
because of the increased computational requirements per iteration,
there is practically no change in the total work required when
compared with algorithms using standard column selection criteria.

(ii) Dual and Primal-Dual Algorithms.

A numerically stable dual method for positive definite quadratic
programs has been developed and fully implemented in FORTRAN.
It uses both Cholesky and QR factorizations. Extensive computational
testing indicates that it is three to four times as efficient as
competitive primal codes. When used as a subroutine in the Harwell

code of Powell's recursive quadratic programming algorithm for
solving nonlinear programming problems a similar speed-up in
performance was observed.

4. Nonlinear Programming Algorithms
(i) Minimization Algorithms which Make Use of Negative Curvature.

Several minimization algorithms which make use of negative curvature
have been developed. These include versions of the Goldfeld-Quandt-
Trotter method and gradient path methods which incorporate negative
curvature information through the numerically stable Bunch-
Parlett factorization.

(ii) Curvilinear Path Steplength Algorithms

A curvilinear path steplength algorithm for use in minimization
methods was developed. In non-convex regions the path followed
is initially tangent to the direction of steepest descent, (or
some other suitable downhill direction), eventually becoming tangent
to a direction of negative curvature. In strictly convex regions
a straight path is followed. Under fairly modest conditions
we have proved that our steplength algorithm ensures convergence
of the iterates generated by the minimization algorithm to a
critical point at which the Hessian of the objective function is
positive semidefinite.

C. PUBLICATIONS:

D. Goldfarb and W. Y. Sit, "Worst Case Behaviour of the Steepest Edge Simplex Method," Discrete Applied Mathematics, 1 (1979) 277-285.

D. Goldfarb, "Curvilinear Steplength Algorithms for Minimization which use Directions of Negative Curvature," Mathematical Programming, 18 (1980) 31-40.

D. Goldfarb and M.J. Todd, "Modifications and Implementation of the Shor-Khachian Algorithm for Linear Programming," Department of Computer Science Technical Report TR 80-406, (1980) Cornell University (also published as School of Operations Research and Industrial Engineering Technical Report No. 446, Cornell University) submitted for publication to Mathematical Programming.

D Goldfarb, "The Use of Negative Curvature in Minimization Algorithms," Department of Computer Science Technical Report TR 80-412 (1980), Cornell University.

R.G. Bland, D. Goldfarb, and M.J. Todd," The Ellipsoid Method: A Survey," School of Operations Research and Industrial Engineering Technical Report No. 476 (1980) Cornell University, submitted for publication to Operations Research.

D. Goldfarb, " Steepest Edge Simplex Algorithms for Network Flow Problems" submitted to Mathematical Programming.

A. Idnani, "Numerically Stable Dual Projection Methods for Solving Positive Definite Quadratic Programs," Ph.D. Thesis, The City University of New York (1980).

D. PARTICIPATING SCIENTIFIC PERSONNEL:

Principal Investigator: Professor Donald Goldfarb

Graduate Research Assistant: Dr. Ashok Idnani (Ph.D. awarded by The City University of New York, May 1980)

FILMED

2-8